Описание алогоритма.

Назовем периодом периодической строки повторяющуюся последовательность символов. Делается попытка найти подстроку минимального размера, которая может считаться кандидатом на то, чтобы быть периодом. Далее выполняется следующий цикл.

1. Если кандидат не найден, строка не периодическая.
2. Если кандидат является периодом, то строка периодичeская, в противном случае делается попытка найти кандидат большего размера, выполняется переход к шагу 1.

При этом учитываются следующие очевидные обстоятельства.

1. Если исходная строка периодическая, то ее размер должен быть кратен размеру периода.
2. Если исходная строка периодическая, то размер периода не может быть больше половины размера исходной строки.

Поиск очередного кандидата выполняет функция candidate(p, s) (s – исходная строка) следующим образом. Находится число n (длина кандидата), удовлетворяющее следующим условиям.

1. Значение n больше p.
2. Значение n не превышает половину размера исходной строки.
3. Подстрока размером n, начинающаяся с начала исходной строки совпадает с подстрокой, начинающейся с символа с номером n.

Проверка того, что найденный кандидат является периодом, выполняет функция isPeriod(p, s) (s – исходная строка) следующим образом.

1. Размер исходной строки кратен p.
2. подстрока размером p, начинающаяся с начала исходной строки, повторяется, начиная с символа с номером 2\*p.

В худшем случае сложность алгоритма имеет порядок O(n2), например, для строки, все символы которой одинаковы, за исключением последнего. В лучшем случае сложность алгоритма имеет порядок O(n), например, для строки, в которой не встречается одинаковых символов (или наоборот, для строки, все символы которой одникаковы). Полагаю, что средняя сложность должна стремиться к O(n), т.к. при равномерном распределении вероятностей доля строк, на которых алгоритм деградирует, сравнительно невелика и должна уменьшаться с ростом n. Ответ на этот вопрос мог бы дать численный эксперимент.